TX-0 COMPUTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE 39, MASSACHUSETTS

M-5001-12-1

March 29, 1960

PREPARATION AND USE OF LIBRARY SUBROUTINES

Properly prepared subroutines in the TX-0 Library can be used to significantly reduce the labor of programming.

1. Use of Subroutine Tapes

Subroutines prepared in the manner described below can be used in either of two ways:

a) The user may copy portions of interest into his own English tape (in which case he omits the "Start" block at the end of the subroutine tape), or

b) The user may compile directly from the subroutine tapes.

As an example of the latter approach, the following sequence of tapes might be used: (The settings TAC=0 and TBR=400020 are to be used throughout.)

ACTION	TAPE CONTENTS
MACRO II in PETR, Read-in	Who knows?
PASS I: Title tape in PETR, Restart	(Title) 1000 | location for subroutines start
Subroutine tape in PETR, Test	Integer Multiply zz=. define intmul M,N . . . . . . start
Subroutine tape in PETR, Test	Decimal Print zz=. define prdec . . . . . . start

etc. . . .

Main Program in PETR.

Test

(Title)
. . .
. . .
. . .
tt. tt+6 | temporary storage
constants
start beg

PASS II: Title tape in PETR.

Restart

Tapes must be processed in the same sequence as Pass I.

Subroutine Tape in PETR. . . .

Test

etc. . . .

Main Program in PETR. . . .

Test

Restart (To punch start block)

MACRO SYMBOL PRINT in PETR (If desired).

Read-in

Tear off binary tape from Flexowriter

MACRO SYMBOL PUNCH IN PETR (If desired).

Read-in

Figure out what went wrong.

The following points should not be overlooked.

a) Macro instructions must be defined before they are used (thus, subroutine tapes containing macros must be processed ahead of the main program).

b) The psuedo-instruction "constants" must appear on some tape following the subroutine tapes, or the constants used by the subroutines will not be provided.

c) Any temporary storage registers used by the subroutines must be reserved. Thus, if the six registers tt through tt+5 are used, the following symbols should appear (only once) on some input tape:

tt. tt+6

2. Preparation of Library Subroutines

Subroutines intended for the TX-0 Library will have greater utility if the following procedures are adhered to.

2.1 Macro-Instructions

A group of instructions that is well suited for use as an open subroutine should be defined as a macro-instruction. For closed subroutines, macro-instructions should be defined for the more useful calling sequences.

In choosing the symbol for a macro, the coder should seek a set of characters that mnemonically identifies the operation with as much precision as possible. For an integer multiplication subroutine, for example, the instruction "intmul" is preferable to "mult", since it may be necessary to use other multiplication subroutines (e.g. fractional or logical) in the same program. In general, six character macros are less likely to conflict than four.

2.2 Addresses

At most, two floating address symbols should be used: zz and tt. zz is used to identify the first register of the subroutine. The value is defined by a parameter assignment (zz=.), which precedes and is used in the definition of the macro-instructions. (In this way, the same symbol, zz, can be used repeatedly in a single assembly without ambiguity.)

For temporary storage, the symbols tt, tt+1, tt+2, etc. should be used. Only quantities that are used in a single pass through the subroutine should be stored in this way. Quantities that must be preserved between successive applications of the subroutine should be allocated storage within the subroutine. (The reservation of the temporary registers tt, tt+1, tt+2, etc. is left to the user of the subroutine.)

If the complexity of the subroutine requires the use of additional address symbols, they should be chosen in a systematic way (e.g., s01, s02, etc.) so that the user can readily check for conflicts.

If the above rules are adhered to, the user will have the following options:

a) He may use the macros and subroutines intact,

b) He may define his own macros to be used with the subroutine,

c) He may use the subroutine without macros. (In this case, he generally must identify the subroutine entry location with a symbol of his own, defined either by a parameter assignment or an address tag.)

2.3 Intelligibility

Comments should be included on the English tape to identify the operations performed by various portions of the subroutine.

Transfers to widely separated locations should be referenced to the subroutine origin (zz) and the location transferred to should be identified by a symbolic address tag. Thus, instead of writing

```
        .
        .
        .
       lac
        .
        .
   (42 instructions)
        .
        .
       trn .-43
```

which involves the reader in a laborious counting operation, the following notation should be used:

```
        .
        .
zz+10,  lac
        .
        .
        .
       trn zz+10
```

2.4 Tape Preparation

In preparing the English tape, the title should be immediately preceded by a bar . There should be at least five inches of tape feed ahead of the title and an inch or two following. The tape should also be advanced between the last macro termination and the beginning of the subroutine.

At the end of each typewritten page, there should be

tape feed (1 inch)

stop code

tape feed (1 inch)

This same sequence should follow the carriage return at the end of the subroutine, and should in turn be followed by

carriage return

"start"

carriage return

tape feed (4 inches or so).

If the above format is followed, the user can readily identify the various segments of the tape.

2.5 Description

The salient properties of the subroutine should be described in a page or two. The following format is suggested.

Subroutine Title - This should be the same as the title at the beginning of the tape (without the bar, if you please).

Function - The purpose of the subroutine should be given as well as the way it should be used. Any limitations (arithmetic or otherwise) should be spelled out.

Space - The number of registers occupied by the subroutine and the number of constants should be given in both octal and decimal forms. The temporary registers used should be specified (e.g., "tt through tt+7").

Time - The time(s) required for an operation of the subroutine should be expressed in terms of relevant parameters. The following instruction times should be used:

trn (with AC-), tra . . . . . 6 μsec.

dis . . . . . 36 μsec.

trn (with AC+), and others . 12 μsec.

Method - A brief description should be given of the method used, including appropriate formulas. Any trickery should be explained, using references to the tape printout. A flow chart is often a convenient form for the exposition.

The name of the author and the date should follow the text of the description.

2.6 Submission

The subroutine write-up should be submitted for filing in the TX-0 Computer Room. One copy each of the English tape, Flexowriter printout, and subroutine description should be included. (The description need not be typewritten.)

An example of the kind of write-up desired is attached.

Author L. D. Earnest
Lester D. Earnest dbh

Approved Jack B Dennis
Jack B. Dennis dbh

LGE/dbh
Attchs.

SUBROUTINE; INTEGER MULTIPLY

1. FUNCTION

The macro-instruction "intmul M,N" will cause the N low order (right-hand) bits of the contents of M to be used as an integral multiplier in the operation

$$C(AC) \times C(M) \longrightarrow C(AC).$$

The product must not exceed eighteen bits.

Both C(AC) and C(M) may be positive or negative. If C(M) will always be positive, faster operation will be obtained by choosing N as small as possible subject to the restriction that C(M) must not exceed 2^N-1. If C(M) can be negative, the value $N=22_8=18_{10}$ must be used.

If either C(AC) or C(M) is +0, the product will be +0. If C(AC) is -0 and C(M) is not +0, the product will be -0. Similarly, the product will be -0 if C(M) is -0 (requiring $N=18_{10}$) and C(AC) is not +0.

2. SPACE

The subroutine occupies $21_{10}=25_8$ registers, uses one constant and temporary storage registers tt, tt+1, and tt+2.

3. TIME

The time required for the calling sequence and subroutine operation is

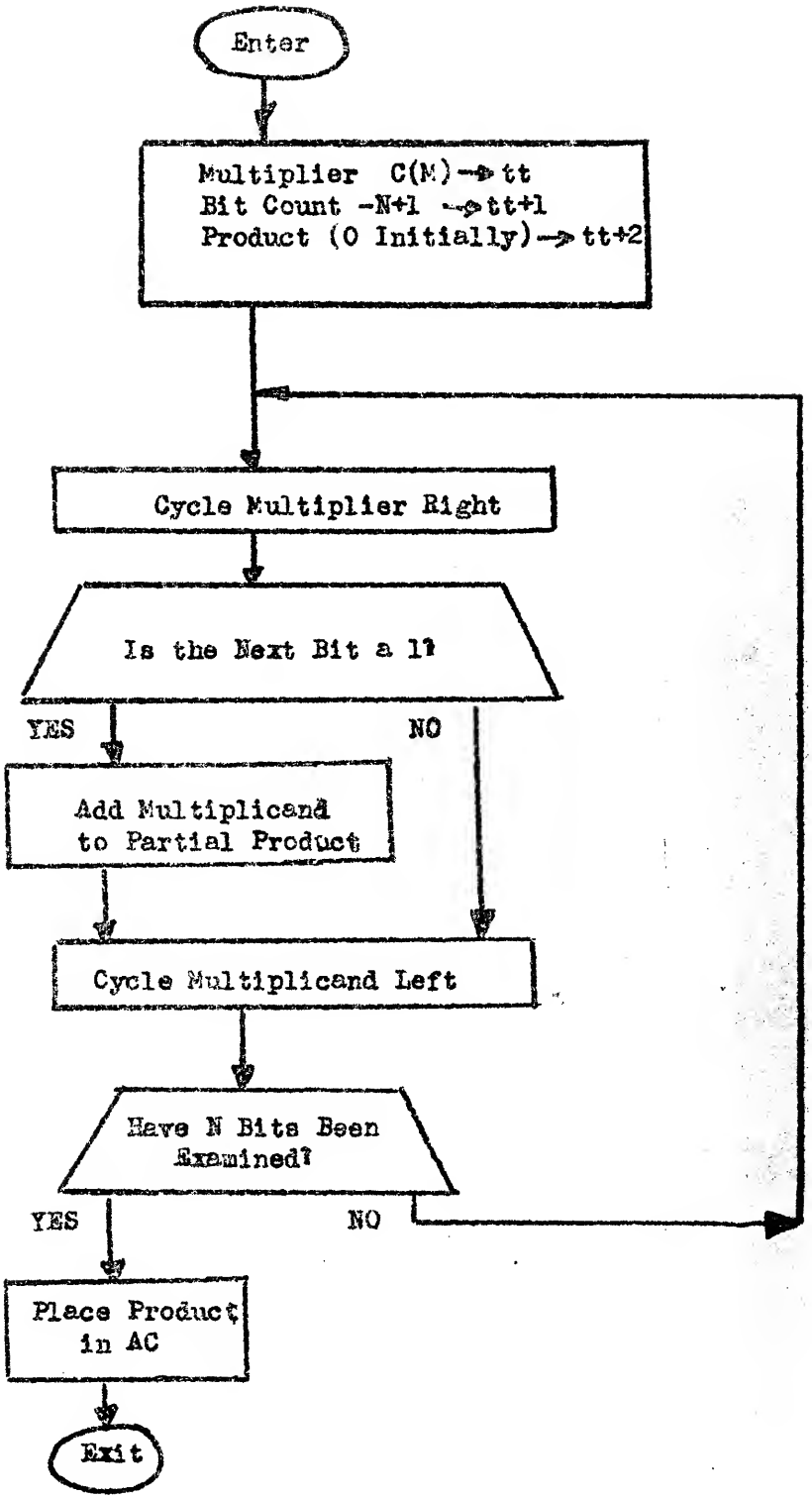
$$0.138 + 0.132N + 0.024b \text{ milliseconds},$$

where b is the number of 1's in the multiplier. Thus, using a full eighteen-bit multiplier, half of whose bits at 1's on the average, the expected operating time is 2.73 msec. Similarly, for a six-bit multiplier, the time is 1.00 msec.

4. METHOD

The operation of the subroutine is described in the accompanying chart.

N. Bonaparte
27 March 1794



INTEGER MULTIPLICATION

```
|INTEGER MULTIPLICATION

zz=.

define   intmul M,N      llr M
                         slr tt
                         llr (-N+1
                         slr tt+1
                         llr (tra .+2
                         tra zz       terminate

|SUBROUTINE

zz,      slr zz+24
         ala
         sto tt+2
zz+3,    cla              |Begin loop.
         add tt
         cyr              |Cycle multiplier right.
         sto tt
         trn .+2          |If there is a 1 in this position, increase product.
         tra .+4
         lac
         add tt+2         |Add multiplicand to partial product.
         sto tt+2
         cla 12           |Cycle multiplicand left.
         ala
         add tt+1
         add (1
         sto tt+1
         trn zz+3         |Go back, if there is more to do.
         cla
         add tt+2
zz+24,   0                |Exit
```